AMENDMENTS TO THE DRAWINGS

The drawings have been objected to because the text is purportedly illegible. Applicant submits herewith appropriately marked Replacement Sheets for Figures 2 and 5-30 that Applicant believes should remedy the above-noted objection to the drawings by providing larger, and hence more legible, text.

REMARKS

This responds to the Final Office Action mailed September 19, 2007, for the above application in which claims 144, 145, 155-158, 160, 164, 165 and 167-172 and new claim 1615 are now pending. Reconsideration of the application and claims in light of the following is requested.

Comments On The "Response to Arguments"

In paragraph 36 of the Office Action, the Patent Office asserts that Applicant's arguments with respect to claim 155 are not persuasive. Applicant disagrees for the reason that the Patent Office is equating "mode" and "type" whereas "mode" connotes a manner of operation, whereas "type" connotes a kind of thing. In context, a single "processor" of Bonola has different "modes" meaning it the same device can operate in different ways depending upon the opcode. In contrast, the instant claims call for three devices – two processing resources of one type, i.e. one kind (the "first type") and another of a different type (the "second type"). In addition, the position of the Office Action seemingly treats the claims in a vacuum, instead of as informed by the Specification of which they are a part. Applicant's Specification makes abundantly clear, at paragraph [0055], what is meant by the types of resources. Accordingly, reconsideration of the position of the Office in paragraph 36 of the Office Action is respectfully requested.

In response to Paragraph 38, it is noted that Applicant did provide argument – namely that "the use of the slave processor's registers to hold data is not an instruction cache, let alone an 'execution-instruction processing cache'" Given that further elaboration is apparently necessary, it is respectfully submitted that the slave registers of Bonola do not constitute an "execution-instruction processing cache" but rather are merely communication buffers that act as a mailbox for a software request from the master microprocessor to the slave microprocessor,

which the latter retrieves when it executes a software interrupt routine triggered by the master writing request data to the slave registers. The slave registers are not a cache and certainly do not execute instructions. A cache is a well understood term with specific meaning in the computer art and simple buffer or queue is not a cache. Moreover, simply executing a software routine is not executing a delegated instruction, it is emulating because there is no because there is no delegation of instructions. Accordingly, reconsideration of the position of the Office in paragraph 38 of the Office Action is respectfully requested.

With respect to paragraph 39 of the Office Action, the Office Action asserts that "to perform a logical operation in a single clock cycle, a system designer need only modify the processing frequency. Such modification is well known and common to a person of ordinary skill in the computer arts . . ." Applicant disagrees with such an assertion. First, the Office Action's position says nothing about how Butterworth could be made to do what was stated. Rather, in effect, it asserts that one could construct a system to perform a logical operation in a single clock cycle. The Office Action assertion is a non sequiter. It is not possible to do so with Butterworth and still have the Butterworth system operate. The reason that Butterworth cannot be modified to perform the claimed method has nothing to do with the frequency, but rather the fact that Butterworth is a software system that takes multiple instructions to complete a transaction. Third, the Office Action offers nothing in support of its blind assertions. If the assertion is true and well known, it should have been easy to provide substantiating evidence of that fact. Accordingly, reconsideration of the position of the Office in paragraph 39 of the Office Action is respectfully requested.

With respect to paragraph 40 of the Office Action, the distinction in meaning between conventional devices and Applicant's use of the terms integer processing unit, mathematical

processing unit or vector processing unit is that they are all coprocessors that can receive individual instructions from a single instruction stream that are delegated to them. The Office Action seems to be requiring a definition akin to "when I use the term X, I mean, ____." Such a definition is not required as a matter of law to define what a term means. The definitions asserted by Applicant are made clear at least paragraphs [0052]-[0057], [0061]-[0063], [0065]-[0068], [0077], [0085], and [0088]-[0090] (as well as their associated figures). From those paragraphs it is abundantly clear that all of the devices are incapable of executing some instructions that can appear in a single instruction stream. In other words, if an instruction stream includes all available instructions, none of the "processing resources" are capable of executing all of the instructions in the stream. Thus, it is necessary have primary instruction executing coprocessors, such as a multiplier, divider and floating point unit, AND, in accordance with the instant claims, the delegated-instruction-executing coprocessors, such as a multiplier, divider and floating point unit, that receive instructions delegated to them. Moreover, a delegated-instruction-executing coprocessor is shared between at least two of the primary instruction executing coprocessors. This approach, clearly set forth in the claims, does not exist in any of the cited references, nor is it even remotely obvious from such references.

Drawing Objection

In the Office Action, figures 2 and 5-30 were objected to as failing to comply with 37 CFR 1.121(d) because the text was allegedly illegible. Appropriately marked "Replacement Sheets" for Figures 2 and 5-30 are attached. Entry of these sheets is requested. It is respectfully submitted that the drawings now comply with 37 CFR 1.121(d).

Section 112 Rejections

In the Office Action, claims 144-145, 155-158, 160, 164-165, 167-172 and 1615 were rejected under 35 U.S.C. 112, first paragraph, as lacking written description. Applicant respectfully traverses the rejection. Applicant has not added any new matter by amendment of the claims in response to the last Office Action. For the convenience of the Examiner, reference is made to the published version of the application(U.S. Pat. Pub. No. 2005/0235134). It is respectfully submitted that all of the claimed subject matter is clearly supported by at least paragraphs [0052]-[0057], [0060]-[0068], [0077], [0085], [0088]-[0090], [0093], [0097], [0100]-[0101], and [0117] along with the associated drawings. All such subject matter was present in the application at the time the application was filed. Accordingly, withdrawal of the rejections of those claims under section 112, first paragraph is respectfully requested.

In paper 20071204, the Office asserted that the response filed November 19, 2007 was not fully responsive because the above citations were a "general reference to large portions of the specification. The position of the Office is respectfully traversed for the reasons stated below. However, in the interest of expediting prosecution, the following supplementation with respect to language added to the rejected claims should be sufficient and deemed fully responsive.

i) the "at least two different types," "first type," "second type" of processing resources are particularly supported in the specification in at least the middle of published paragraph [0054] where multiple different types of exemplary processing resources are identified:

> Processing resources may include digital signal processing units (DSPs), graphic processing units (GPUs), IPUs, input/output controller processing units, memory management units (MMUs), MPUs, processing cache memory, vector processing units (VPUs), and/or the like. For example, in certain embodiments such as 3D

> animation rendering, floating-point throughput is desirable and Processor 100 may be configured with a single IPU and four MPUs interconnected by a Router. In yet another example embodiment, such as for a high traffic web server, handling multiple web requests simultaneously may require Processor 100 configured with 16 IPUs, a single MPU, and larger processing cache interconnected by a Router.

ii) the "second type . . . shared by at least two processing resources of the first type" is specifically supported by at least published paragraph [0053] which states:

> In this exemplary architecture, four IPUs 102, 104, 106, 108 are collaborating with single MPU 112 and L2 cache 114 (i.e., a single MPU 112 and L2 cache 114 are shared by four IPUs 102, 104, 106, 108).

iii) the "executing the first individual thread" is specifically supported by at least paragraphs [0056]-[0057] which state, in pertinent part:

> IPUs 102, 104, 106, 108 are the general integer and logic processing units of Processor 100 and each of the IPUs is responsible for running a separate program thread (i.e., a part of a program that can be executed independently of other IPUs).
>
> * * *
>
> Each of IPUs 102, 104, 106, 108 is configured to execute the instructions of a thread and/or process and perform relatively simple calculations of the instructions such as add, subtract, basic logic (e.g., Boolean operations) and/or integer calculations. Each of the IPUs are further configured, upon decoding an instruction while executing the instructions of the thread and/or process, to send calculation requests along with data and opcode of the instruction to MPU 114 for relatively complex calculations such as multiplication, division and/or floating point calculations. Alternatively, the IPUs may be configured to send the instructions themselves that require complex calculations to the MPU without decoding the instructions. Similarly, each of the IPUs may also be configured, while executing the instructions of the thread and/or process, to send access requests to L2 cache 112 for accessing data and/or instructions stored in the L2 cache.

iv) the "execution instruction" is specifically supported by at least paragraph [0055] which, in pertinent part, states:

> Further, through instruction set design and intra-chip networking via a Router, these intra-dependent processing resources are capable of identifying other processing resources to which they may delegate instruction signals that they themselves may be incapable of servicing.

v) the "execution instruction signal" is specifically supported by at least paragraph [0097] which states:

> In particular, after a requesting IPU (e.g., IPU 102) provides a request to another processing resource (e.g., L2 cache 112 and/or MPU 114) independently, the requesting IPU may go to "sleep" until a responding processing resource (e.g., L2 cache 112 and/or MPU 114) returns the requested information. Sleep is an important capability of Processor 100. Sleep may be induced by a switch unit in an embodiment, which may be placed between a power or clock providing line-in to any processing resource. Upon instruction, the switch unit may act to switch off power or clock to the unit while waiting for a response thereby reducing the overall power requirements for Processor 100. The resulting outputs from a sleeping processing resource may be configured to be non-existing, and as such they do not interfere with other competing inputs. For example, if IPU 102 or MPU 114 is put to sleep, their outputs 116, 142 and 148 may not affect and are no longer interpreted as inputs at their destinations. The switch unit may be switched on by other processing units, typically, upon completion of a request at a shared resource (e.g., MPU 114) and/or freeing of a processing resource (e.g., a locked memory location being unlocked in L2 cache 112). Alternatively, various processing resources may be shut off upon instructional direction so as to save power. For example, in an environment where conserving power is desirable (e.g., a battery powered laptop computer), a program (e.g., an operating system) may instruct the processor to shut off various processing resources to extend battery longevity. In yet another alternative embodiment, such a program may dynamically turn on and off various processing resources to maintain a desired level of power draw while maximizing processing throughput In yet another embodiment, Processor 100 itself shuts off processing resources while idling. Alternatively, the delegating processing

> resource (e.g., IPUs 102, 104, 106, 108) may continue to operate when requested for concurrent processes.

Finally, claim 1615 is specifically supported by the above same specific paragraphs as well as in the overall paragraphs identified in the November 19, 2007 response. It is impossible to more narrowly limit the identification of support for this claim, and the assertion in paper 20071204 is traversed, because the application description is at a fine degree of granularity, including numerous variations along the way, whereas the claim is more general.

It is now respectfully submitted that, to the extent the November 19, 2007 response was asserted to be not fully responsive, this supplementation eliminates the issue.

In addition, claims 144, 155-156, 164-165, 167-170 and 1615 were rejected under 35 U.S.C. 112, second paragraph, because the term "the processing resource of the first type" allegedly lacks antecedent basis. Applicant respectfully traverses the rejection. Claim 144 as amended in the prior Response also amended the claim to recite "obtaining an execution instruction, wherein the execution instruction is obtained at one of at least two processing resources of a first type; . . ." (emphasis added herein). Accordingly, it is respectfully submitted that the Final Office Action is in error and the claims are all definite. Accordingly, withdrawal of the rejections of those claims under section 112, second paragraph is respectfully requested.

Art Rejections

In the Office Action, claims 144, 155-156, 164-165 167-170 and 1615 were rejected under 35 U.S.C. 103(a) as being anticipated by Bonola, U.S. Pat. No. 5,706,514 (Bonola) in view of Bridges et al., U.S. Pat. No. 6,081,860 (Bridges). Applicant respectfully disagrees.

As noted in response to the prior Office Action, Bonola discloses a conventional multiprocessor configuration in which each of the processors executes its own thread from

memory but are interconnected in a form of master-slave relationship. When a particular processor of Bonola executes its thread it is a master. When it receives delegated instructions (because of a mode mismatch or busy condition) it is a slave. In all cases, the various processors are all complete processors that may run in different modes.

Similarly, Bridges describes a multiprocessor system consisting of multiple full microprocessors that connect to multiple memory devices via a shared, arbitrated Processor Local Bus (PLB). The arbiter accepts a memory request while another is in progress, pipelining the address of the next request so data transfer can begin immediately after the conclusion of the previous request. Bridges refers to the microprocessors as "master devices" and the memory devices as "slave devices", but it is a memory address that is pipelined and memory data that is transferred so, properly read in context, the meaning is clear.

In contrast, claim 144 recites "obtaining an execution instruction, wherein the execution instruction is obtained at one of at least two processing resources of a first type" and "determining whether an operation-code within the execution instruction should be delegated to an other processing resource of a second type different from the first type and shared by at least two processing resources of the first type" which is different and distinguishes over Bonola in combination with Bridges because the memory of Bridges is not a processing resource as that term is used in the instant application. Therefore, no combination of Bridges with Bonola would meet or render claim 144 obvious because they both would still lack the processing resource of the second type that is shared by at least two processing resources of a first type and the second type is different from the first type. Accordingly, claim 144 should be allowed.

All of the remaining pending claims (except claim 1615) depend, directly or indirectly from claim 144 and are thus allowable for the same reason.

Moreover, it is respectfully noted that the rejections of claims 155, 156, 164-170 and 1615 make no mention of Bridges. Applicant previously identified the relevant distinctions between Bonola and the instant claims. Bridges adds noting to remedy that lack of disclosure. Accordingly, since applicant is left to guess why Bridges is relevant to these dependent claims. Withdrawal of the rejections of claims 155, 156, 164-170 and 1615 is therefore respectfully requested for this independent reason.

Still further, the dependent claims add aspects that independently distinguish over Bonola. For example, claim 155 expressly recites that "the operation-code indicates a type of resource on which to execute." There is no such disclosure in Bonola alone or in combination with Bridges – Bonola discloses different modes and different modes are not the same as different types of resources. Bonola does not disclose a operation code that indicates a "type of resource on which to execute." Bridges does nothing to remedy the deficiency of Bonola.

Claim 164 expressly recites that the processing resource of the second type is an execution-instruction processing cache. Contrary to the assertion in the Office Action, the use of the slave processor's registers to hold data is not an instruction cache, let alone an "execution-instruction processing cache" as recited in that claim. No combination of Bridges and Bonola would meet that limitation. Consequently, amended claim 164 is allowable.

With respect to claim 165, Bonola describes a host processor passing parameters and a code starting address to one or more slave processors, either via registers [col. 3, lines 37-42], or shared memory [col. 8, lines 12-17; col. 9, lines 49-52] on a host bus [col. 4, lines 43-45]. This has nothing to do with the instant claims embodiments of which, for example, would be set up with multiple processors sharing a single coprocessor or multiple coprocessors each of different hardware functionality [figs. 2, 12; 0053, 0059], requiring an instruction-instruction signal router

such as a hardware multiplexor or crosspoint switch with hardware arbitration connecting them directly, not indirectly via shared memory. No such subject matter is even remotely disclosed by Bonola or Bridges, alone or in combination. Claim 165 is allowable.

In addition, claims 144, 171-172 and 1615 were rejected as anticipated by under 35 U.S.C. 102(e) as being anticipated by Butterworth et al., U.S. Pat. No. 6,907,454 (Butterworth) in view of Bridges. Applicant respectfully asserts that such a rejection is improper, contrary to law and prejudicial. It is respectfully submitted that an anticipation rejection must be based upon a single reference, not a combination of references. Accordingly, withdrawal fo the rejection is respectfully requested.

Notwithstanding the impropriety of the rejection, no combination of Bridges with Butterworth would result in what is claimed in those claims. As previously noted, like Bonola, Butterworth also discloses a multiprocessor master-slave arrangement in which the master processor is an IBM Power PC 705 and the slave is a Power PC 403 microprocessor. In other words, like Bonola, in Butterworth both the master and slave processors are complete microprocessors that execute independent instruction threads from memory. Accordingly, all of the arguments made above with respect to Bonola apply with equal force to Butterworth. Still further, Bridges remedies nothing deficient from Butterworth. Moreover, as noted above, Bridges does not disclose slave processors, the "slave" elements of Bridges are memory. Accordingly, claim 144 is not anticipated (or obvious), and hence patentable over, Butterworth (or a combination of Butterwort with Bridges) for the same reasons.

Claims 171 and 172 each depend from claim 144 and are patentable for the same reasons.

In addition, claim 145 was rejected under 35 U.S.C. 103(a) as being unpatentable for obviousness over Butterworth in combination with Bridges. Applicant respectfully disagrees.

First, it is respectfully noted that, although Applicant distinguished Butterworth in the prior Response, the Office Action cites to nothing in Bridges that adds to the disclosure relevant of Butterworth. Accordingly, it is respectfully submitted that the rejection is an improper attempt to maintain a deficient rejection simply be referencing another patent. If the Office Action wanted to assert that Bridges added relevant disclosure it should have said so.

The Office Action argues that it would have been obvious to "complete the instruction-delegation method disclosed in Butterworth in a single cycle." From a substantive point of view, it is irrelevant if Butterworth can post a memory request to the slave microprocessor in one cycle. The instant claim relates to delegating an instruction to a shared coprocessor in one cycle. See paragraph [0066]. Butterworth's slave microprocessor executes software from memory and takes many cycles to complete a transaction, whereas the same is not true for the invention of this claim in which the transaction would be completed in one or a few cycles. The Office Action tacitly acknowledges that Butterworth would not perform the method of claim 144 in a single cycle. Moreover, it is doubtful that Butterworth could even be modified to do so without a wholesale redesign of that system – a nontrivial task to say the least. It is improper to reject a claim for obviousness over prior art where, to meet the claim, the prior art would have to be modified in nonspecific and unspecified significant ways. The rejection should be withdrawn.

If the Patent Office maintains the rejection of claim 145 on this ground, it is respectfully requested that the Office identify exactly how one would have purportedly modified Butterworth in a way that would have achieved the claimed invention and how the Office knows that such a modification to Butterworth would still work.

Claims 157-158 and 160 were rejected for obviousness over Bonola in further view Bridges and Mohamed et al., U.S. Pat. No. 5,978,838 (Mohamed). Applicant disagrees.

Claims 157-158 and 160 all ultimately depend from allowable claim 144. Accordingly, those claims are allowable for the same reasons.

In addition, with respect to those claims, the respective "integer processing unit," "mathematical processing unit" and "vector processing unit" are all specifically defined in the instant application, as applicant is entitled to do. They are not what might be conventionally referred to by those terms. See, for example, the Application at p. 12 and the paragraphs cited above. Thus, while the terminology may be the same, the expressly recited components are completely different in Mohamed. Mohammed describes a multiprocessor system in which a host or control processor connects to and controls a slave processor [col. 3 lines 2-5] in a manner similar to Bonola, except that the slave processor is specialized to the execution of vector SIMD software [fig. 2; col. 4 lines 33-34]. Mohammed refers to the slave microprocessor as a "co-processor", but this word does not have the same meaning as used in the instant application where it cannot mean a processor that executes its own software from memory [col. 4 lines 23-25]. Mohammed specifically discusses "coprocessors" which are akin to applicant's processors but Mohammed's are indisputably unshared [col. 1, lines 19-30]. Mohammed distinguishes his use of the word [col. 1, lines 31-33; col 1, lines 61-67]. None of the cited art discloses, teaches or remotely suggests components such as the "integer processing unit," "mathematical processing unit" and "vector processing unit" as those terms are used in the instant application. Accordingly, the alleged combination of Bonola and Mohamed would not achieve or render the claimed invention obvious.

CONCLUSION

It is respectfully submitted that all of the pending claims are allowable and early favorable action in that regard is solicited. In addition, this Response is being submitted within

two months of the mailing of the Final Office Action. In the event any issues remain that could potentially be resolved by telephone, the Examiner is urged to contact the undersigned at the number indicated below. Although no fees are believed to be due, should any additional fee(s) be required for the entry of this Amendment, the Commissioner is hereby authorized to charge Deposit Account No. **13-4500**, Order No. **4403-4000US6**.

Respectfully submitted,

MORGAN & FINNEGAN

Dated: December 19, 2007 By: 

Richard Straussman
Reg. No. 39,847

CORRESPONDENCE ADDRESS:
Correspondence Address associated with
Customer Number: **27123**